

# IP Multicast Addresses

## *Catalog of those in use for IRMs*

Wed, Jun 18, 1997

A range of IP multicast addresses was assigned by networking for use by IRMs here at Fermilab. They are 239.128.02.xxx. The software in IRMs supports up to 15 of these, plus broadcast. They are selected for transmission by node#s 09Fx, where 09FF always means broadcast as used for ARP requests. This table shows the current usage of such addresses:

09F0	PET alarms
09F1	
09F2	A0 alarms
09F3	
09F4	
09F5	
09F6	
09F7	
09F8	
09F9	All IRM nodes
09FA	All PET nodes
09FB	All BRF nodes (Booster HLRF)
09FC	All A0 nodes (Tesla/Photo-Injector)
09FD	
09FE	
09FF	(broadcast)

In the non-volatile memory of IRMs there is a table of 16 entries that corresponds to the above list. Each 8-byte entry contains a multicast hardware address and a diagnostic count of the number of frames transmitted to that destination multicast address. As an example, here is the table from a PET node:

0584:405B80	0100	5E00	02F0	0000
0584:405B88	0000	0000	0000	0000
0584:405B90	0000	0000	0000	0000
0584:405B98	0000	0000	0000	0000
0584:405BA0	0000	0000	0000	0000
0584:405BA8	0000	0000	0000	0000
0584:405BB0	0000	0000	0000	0000
0584:405BB8	0000	0000	0000	0000
0584:405BC0	0000	0000	0000	0000
0584:405BC8	0100	5E00	02F9	0000
0584:405BD0	0100	5E00	02FA	0000
0584:405BD8	0000	0000	0000	0000
0584:405BE0	0000	0000	0000	0000
0584:405BE8	0000	0000	0000	0000
0584:405BF0	0000	0000	0000	0000
0584:405BF8	FFFF	FFFF	FFFF	038B

The ethernet multicast addresses used conform to the ethernet convention for IP multicast. The upper 25 bits are fixed to the value 0100 5E as shown. The lower 23 bits match the lower 23 bits of the corresponding IP multicast address. In our case, this is so. As a specific example, this node can access all other PET nodes by

The IP multicast (Class D) address used would be 239.128.2.250, or \$EF8002FA. In order for a node to be able to use a given IP multicast address for *transmission*, the corresponding ethernet multicast address must be installed in the above table.

For a node to be able to *receive* from a given multicast address, the address must be in a different table that has room for up to 8 different ethernet multicast addresses. Looking at the same node used in the above example, we have:

```
0584:405240 0100 5E00 0001 00BB
0584:405248 0100 5E00 02F9 00D4
0584:405250 0100 5E00 02FA 00C7
0584:405258 0000 0000 0000 0000
0584:405260 0000 0000 0000 0000
0584:405268 0000 0000 0000 0000
0584:405270 0000 0000 0000 0000
0584:405278 0000 0000 0000 0000
```

In order to participate in the IGMP protocol that is used by multicast routers to determine whether to pass multicast IP datagrams onto its connected networks, each node must listen for the "all hosts" multicast address 224.0.0.1, which uses the ethernet address 0100 5E00 0001 in the first entry of this table. The last two bytes in each 8-byte entry are a delay counter byte and a diagnostic counter that counts the reception of frames addressed to the given ethernet multicast address. Once each minute, the multicast router sends an IGMP request message to the "all hosts" address that asks the question, "what IP multicast addresses are of interest for the nodes on this network?" Each node that receives this request schedules an IGMP reply message to be sent after a random delay over the following 10 seconds. This is done for each multicast address in this table (but not for the "all hosts" address). The delay counter byte is set to a random value for each entry in use. For IRMs, the delay function is provided by decrementing this byte each 15 Hz cycle. If the counter byte reaches zero, an IGMP reply is sent to the same multicast address. (The router will see this because it can see all multicast frames.) Because the reply is targeted to the same multicast address in which the node is announcing its interest, any other node on the connected network that has the same interest will also receive the reply message; as it does so, it will cancel its own intent to transmit the same reply. In this way, minimal network traffic is required to keep the router up-to-date on which multicast addresses are needed on that connected network. (The router only needs to know if *any* node on a connected network has an interest in a given multicast address. It doesn't care which ones or even how many such nodes there are. One node's interest is enough to force the router to pass such addressed IP datagrams.)

Entries to be added to the above non-volatile table of multicast addresses enabled for reception are made manually as a part of system configuration. Changes to the

table may be made during system operation, and such changes will become effective right away. The system maintains a checksum of the multicast addresses in the table that it calculates about once a second. If it notices a change in the computed checksum value, it builds a new Multicast Setup command and sends it to the ethernet controller chip, an Intel 82596 on the MVME162 board. When it does so, it also sends a diagnostic Dump command that causes the chip to write a 300-byte record of various internal information it keeps. Within this block can be found the contents of the 64-bit hash register that is used for filtering multicast-addressed frames that the controller chip sees on the network. As soon as the chip detects a destination multicast address at the start of an ethernet frame, it hashes the 48-bit address into a 6-bit value. If the corresponding bit in the hash register is set, the frame is accepted, else it is rejected. Because this scheme permits reception of multicast addresses that may only coincidentally hash to the same 6-bit code of another that is in use, the network software must check that a received multicast frame is really addressed to a multicast address of interest, ignoring it if it is not. To that end, the above table is scanned by the ethernet receive interrupt software for multicast-addressed frames. But in order to derive maximum benefit of chip-level filtering of multicast frames, one may prefer to select multicast addresses to be used in a given installation so that they do *not* correspond to matching 6-bit hash codes. To do this, it is useful to examine the hash register contents in the 300-byte Dump area at \$1600C0. Here is the beginning of that area in the same node above:

```
0584:1600C0 7F2E 0060 00F2 0000
0584:1600C8 40FF 003F FFFF 0240
0584:1600D0 0000 0584 0020 83B7
0584:1600D8 3117 9A3F C228 0000
0584:1600E0 0400 42A6 1220 1000
0584:1600E8 0000 0000 1100 D586
0584:1600F0 05FC 0C00 FFFF FFFF
0584:1600F8 0000 0000 00C0 0016
```

The hash table register is located at offset \$26, where we find 1000 0000 0000 1100. Thus, the register has three bits set, so that each multicast address uses a different hash code.

Specification of common usage of multicast addresses in IRMs is done via two node#s in yet another non-volatile table. The first is the "broadcast" node# that is used when an IRM must perform a device name lookup when it cannot find the name in its own device tables. This same node# is also used when an IRM must send a data request that is to be targeted to more than one other node. (Using multicast for this means that only a single message need be sent, rather than one to each node represented in the request message.) The second node# that may be specified is the target node# for Classic protocol alarm messages. Again, as an example, here is what is in that table for the above PET node:

0584:402000 0A00 D400 09FA 09F0

The "broadcast" node#, at \$402004, is 09FA.

The alarms target node#, at \$402006, is 09F0.

By configuring sets of nodes to use different multicast addresses, we can permit each set of nodes to operate in its own network world. This means that IRMs from different sets may use the same 60-character device names, for example. It also means that when the "broadcast" node# is used to send requests for devices from multiple other nodes, such requests will be seen only by the nodes within a set. For example, PET device names could match Booster device names, if desired. A node in the A0 set would not be constrained to use device names that are neither used in Booster HLRF nor in PET. And PET alarm messages received by an alarm handler will not have to filter out Booster HLRF alarm messages. IRMs have found to be generally useful front ends that can serve in various projects, even independent of Acnet. Some of these front ends are ultimately shipped to different labs even in other countries.